


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

longword instructions first second

THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

 Terms used **longword instructions first second**

Found 95,887 of 156,259

 Sort
results
by

relevance


[Save results to a Binder](#)
[Try an Advanced Search](#)
[Try this search in The ACM Guide](#)

 Display
results

expanded form


[Search Tips](#)
☐ Open results in a new window

Results 1 - 20 of 200

 Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale

1 [The instruction parsing microarchitecture of the CVAX microprocessor](#)

David W. Archer

December 1987

Proceedings of the 20th annual workshop on Microprogramming

Full text available: pdf(682.89 KB)

 Additional Information: [full citation](#), [abstract](#), [references](#)

CVAX is a single chip, CMOS VLSI VAX microprocessor. Several microarchitectural innovations helped achieve the desired performance goal of this machine. In particular, the instruction parsing and prefetching mechanism is different from other VAX implementations. This new instruction parsing microarchitecture is discussed in this paper.

2 [Using registers to optimize cross-domain call performance](#)

Paul A. Karger

April 1989

ACM SIGARCH Computer Architecture News , Proceedings of the third international conference on Architectural support for programming languages and operating systems, Volume 17 Issue 2

Full text available: pdf(1.30 MB)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

This paper describes a new technique to improve the performance of cross-domain calls and returns in a capability-based computer system. Using register optimization information obtained from the compiler, a trusted linker can minimize the number of registers that must be saved, restored, or cleared when changing from one protection domain to another. The size of the performance gain depends on the level of trust between the calling and called protection domains. The paper presents alternate ...

3 [A case study of VAX-11 instruction set usage for compiler execution](#)

Cheryl A. Wiecek

March 1982

Proceedings of the first international symposium on Architectural support for programming languages and operating systems, Volume 10 , 17 Issue 2 , 4

Full text available: pdf(642.77 KB)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Analysis of an instruction set as large and varied as the one specified for the VAX-11 architecture is important for aiding processor design evaluation. This paper looks at dynamic VAX-11 instruction set usage by one class of programs, and discusses the methodology and tools which have been developed to provide that information. Six VAX/VMS native mode compilers from Digital Equipment Corporation were used: BASIC, BLISS, COBOL, FORTRAN, PASCAL, and PL/I. A summary of results generated by an ...

4 [Alpha AXP architecture](#)

Richard L. Sites

February 1993

Communications of the ACM, Volume 36 Issue 2

Full text available: pdf(4.62 MB)

 Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#), [review](#)
Keywords: Alpha AXP chip

5 Measuring VAX 8800 performance with a histogram hardware monitor

D. W. Clark, P. J. Bannon, J. B. Keller

May 1988

ACM SIGARCH Computer Architecture News , Proceedings of the 15th Annual International Symposium on Computer architecture, Volume 16 Issue 2Full text available:  [pdf\(1.04 MB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper reports the results of a study of VAX 8800 processor performance using a hardware monitor that collects histograms of the processor's micro-PC and memory bus status. The monitor keeps a count of all machine cycles executed at each micro-PC location, as well as counting all occurrences of each bus transaction. It can measure a running system without interfering with it, and this paper's results are based on measurements of live timesharing. Because the 8800 is a microcoded machine ...

6 The Clipper processor: instruction set architecture and implementation

W. Hollingsworth, H. Sachs, A. J. Smith

February 1989 **Communications of the ACM**, Volume 32 Issue 2Full text available:  [pdf\(4.67 MB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Intergraph's CLIPPER microprocessor is a high performance, three chip module that implements a new instruction set architecture designed for convenient programmability, broad functionality, and easy future expansion.

7 Cache Performance in the VAX-11/780


Douglas W. Clark

February 1983 **ACM Transactions on Computer Systems (TOCS)**, Volume 1 Issue 1Full text available:  [pdf\(880.15 KB\)](#)Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

Keywords: hardware monitor, hit ratio

8 Performance of the VAX-11/780 translation buffer: simulation and measurement

Douglas W. Clark, Joel S. Emer

February 1985 **ACM Transactions on Computer Systems (TOCS)**, Volume 3 Issue 1Full text available:  [pdf\(2.36 MB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

A virtual-address translation buffer (TB) is a hardware cache of recently used virtual-to-physical address mappings. The authors present the results of a set of measurements and simulations of translation buffer performance in the VAX-11/780. Two different hardware monitors were attached to VAX-11/780 computers, and translation buffer behavior was measured. Measurements were made under normal time-sharing use and while running reproducible synthetic time-sharing work loads. Reported measure ...

9 Pipelining and performance in the VAX 8800 processor

Douglas W. Clark

October 1987

Proceedings of the second international conference on Architectural support for programming languages and operating systems, Volume 15 , 22 , 21 Issue 5 , 10 , 4Full text available:  [pdf\(531.67 KB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The VAX 8800 family (models 8800, 8700, 8550), currently the fastest computers in the VAX product line, achieve their speed through a combination of fast cycle time and deep pipelining. Rather than pipeline highly variable VAX instructions as such, the 8800 design pipelines uniform microinstructions whose addresses are generated by instruction unit hardware. This design approach helps achieve a fast cycle time, which is the prime determinant of performance. Some preliminary measurements of cycles ...

10 The rewards of generating true 32-bit code

Michael Franz


January 1991 **ACM SIGPLAN Notices**, Volume 26 Issue 1Full text available:  [pdf\(207.80 KB\)](#)Additional Information: [full citation](#), [index terms](#)

11 Shasta: a low overhead, software-only approach for supporting fine-grain shared memory

Daniel J. Scales, Kourosh Gharachorloo, Chandramohan A. Thekkath

October 1996

Proceedings of the seventh international conference on Architectural support for programming languages and operating systems, Volume 30 , 31 Issue 5 , 9

Full text available:  pdf(1.49 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper describes Shasta, a system that supports a shared address space in software on clusters of computers with physically distributed memory. A unique aspect of Shasta compared to most other software distributed shared memory systems is that shared data can be kept coherent at a fine granularity. In addition, the system allows the coherence granularity to vary across different shared data structures in a single application. Shasta implements the shared address space by transparently rewriting ...

12 The KScalar simulator

J. C. Moure, Dolores I. Rexachs, Emilio Luque

March 2002

Journal on Educational Resources in Computing (JERIC), Volume 2 Issue 1

Full text available:  pdf(493.35 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Modern processors increase their performance with complex microarchitectural mechanisms, which makes them more and more difficult to understand and evaluate. KScalar is a graphical simulation tool that facilitates the study of such processors. It allows students to analyze the performance behavior of a wide range of processor microarchitectures: from a very simple in-order, scalar pipeline, to a detailed out-of-order, superscalar pipeline with non-blocking caches, speculative execution, and comp ...


Keywords: Education, pipelined processor simulator

13 Efficient decomposition and performance of parallel PDE, FFT, Monte Carlo simulations, simplex, and sparse solvers

Zarka Cvetanovic, Edward G. Freedman, Charles Nofsinger

November 1990

Proceedings of the 1990 ACM/IEEE conference on Supercomputing

Full text available:  pdf(1.07 MB)

Additional Information: [full citation](#), [abstract](#), [references](#)


In this paper, we describe the decomposition of six algorithms: two Partial Differential Equations (PDE) solvers (*Successive Over-Relaxation* (SOR) and *Alternating Direction Implicit* (ADI)), Fast Fourier Transform (FFT), Monte Carlo simulations, Simplex linear programming, and Sparse solvers. The algorithms were selected not only because of their importance in scientific applications, but also because they represent a variety of computational (structured to irregular) and communication ...

14 Porting OpenVMS from VAX to Alpha AXP

Nancy Kronenberg, Thomas R. Benson, Wayne M. Cardoza, Ravindran Jagannathan, Benjamin J. Thomas

February 1993

Communications of the ACM, Volume 36 Issue 2

Full text available:  pdf(3.29 MB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

Keywords: 64-bit address space, Alpha AXP, I/O mailboxes, OpenVMS, RISC, VMS, atomic instructions, memory granularity, memory read/write ordering, operating system porting, page tables, synchronized access to shared data, translation buffers

15 Classification and performance evaluation of instruction buffering techniques

Lizyamma Kurian, Paul T. Hulina, Lee D. Coraor, Dhamir N. Mannai

April 1991

ACM SIGARCH Computer Architecture News , Proceedings of the 18th annual international symposium on Computer architecture, Volume 19 Issue 3

Full text available:  pdf(940.03 KB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

16 Multiprocessor cache analysis using ATUM

R. L. Sites, A. Agarwal


May 1988

ACM SIGARCH Computer Architecture News , Proceedings of the 15th Annual International Symposium on Computer architecture, Volume 16 Issue 2Full text available:  pdf(1.38 MB)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The design of high-performance multiprocessor systems necessitates a careful analysis of the memory system performance of parallel programs. Lacking multiprocessor address traces, previous multiprocessor performance studies using analytical models had to make an inordinate number of assumptions about the underlying memory reference patterns. We previously developed a scheme called ATUM - Address Tracing Using Microcode - to get reliable operating system and multiprogramming traces on single ...

17 A characterization of processor performance in the VAX-11/780

Joel S. Emer, Douglas W. Clark

August 1998 **25 years of the international symposia on Computer architecture (selected papers)**Full text available:  pdf(1.12 MB)Additional Information: [full citation](#), [references](#), [index terms](#)**18 A Characterization of Processor Performance in the vax-11/780**

Joel S. Emer, Douglas W. Clark

January 1984 **ACM SIGARCH Computer Architecture News , Proceedings of the 11th annual international symposium on Computer architecture, Volume 12 Issue 3**Full text available:  pdf(980.44 KB)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper reports the results of a study of VAX-11/780 processor performance using a novel hardware monitoring technique. A micro-PC histogram monitor was built for these measurements. It keeps a count of the number of microcode cycles executed at each microcode location. Measurement experiments were performed on live timesharing workloads as well as on synthetic workloads of several types. The histogram counts allow the calculation of the frequency of various architectural events, such as ...

19 On tuning the microarchitecture of an HPS implementation of the VAX


James E. Wilson, Steve Melvin, Michael Shebanow, Wen-mei Hwu, Yale N. Patt

December 1987 **Proceedings of the 20th annual workshop on Microprogramming**Full text available:  pdf(744.71 KB)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

The HPS Microarchitecture has been developed as an execution model for implementing various architectures at very high performance. A considerable amount of effort has gone into the use of HPS as a microarchitecture for the VAX. In this paper, we describe our first full simulation of the microVAX subset, and report the results of varying (i.e. tuning) certain important parameters.

20 The suitability of the VAX for a course in assembly language

Robert W. Sebesta

February 1983 **ACM SIGCSE Bulletin , Proceedings of the fourteenth SIGCSE technical symposium on Computer science education, Volume 15 Issue 1**Full text available:  pdf(423.55 KB)Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

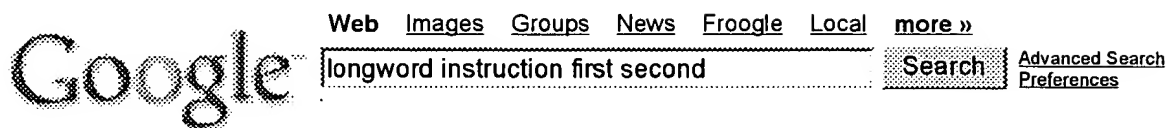
This paper describes the assembly language course we teach, using a Digital Equipment Corporation VAX-11/780 minicomputer, in which structured programming is stressed. It also discusses the relative merits and disadvantages of choosing the VAX as the computer to be used in such a course. The first section of the paper provides a quick survey of the VAX architecture. The second describes our course in assembly language, including our method of structuring assembly language program ...

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)



Web

Results 1 - 10 of about 42,000 for longword instruction first second . (1.03 seconds)Did you mean: **longsword** instruction first secondPatent 4897779: Method and apparatus for optimizing inter ...thereafter, transmitting from the **second** processor to the **first** processor a signal... In the case of a three-**longword instruction**, a complete transfer is ...www.freepatentsonline.com/4897779.html - 35k - [Cached](#) - [Similar pages](#)Patent 4825355: Instruction format for program control type data in response to an **instruction** having an operation code, **first** and **second** ...In response to the decode operation, the **first instruction** of a sequence of ...www.freepatentsonline.com/4825355.html - 33k - [Cached](#) - [Similar pages](#)

[More results from www.freepatentsonline.com]

Logical Instructions in Assembly LanguageSome processors have an **instruction** that sets or clears a bit or byte in ...bit mask (**first** operand, register or memory) with the source (**second** operand, ...www.osdata.com/topic/language/asm/logicop.htm - 21k - [Cached](#) - [Similar pages](#)Advanced Math Instructions in Assembly Language... two data registers contain the byte, word, or **longword** table entries (a ...**first** data register-based table entry is subtracted from the **second** data ...www.osdata.com/topic/language/asm/advmath.htm - 23k - [Cached](#) - [Similar pages](#)VAX MACRO and Instruction Set Reference ManualAfter the interruption, the **instruction** resumes transparently. ... N|| <--- {**first**byte} LSS {**second** byte};. Z|| <--- {**first** byte} EQL {**second** byte}; ...www.openvms.compaq.com:8000/72final/4515/4515pro_025.html - 40k - [Cached](#) - [Similar pages](#)The Sega Programming NetworkAn add **instruction** simply adds the **first** operand to the **second** operand. ...After completing that **instruction**, it reads the next **longword**, and copies that. ...www.emulationzone.org/projects/metalix/megas3.htm - 14k - [Cached](#) - [Similar pages](#)Lecture 18... The result of the **first instruction** is needed to begin the work of the **second**.... Anti-dependency: The **second instruction** overwrites the variable whose ...www.cs.albany.edu/~sdc/csi504/Lect18/ - 7k - [Cached](#) - [Similar pages](#)68000 Instructions - Chapter 4 - AssemblersThis **instruction first** checks to see if the condition is false. ... sized signedoperands and leaves a **long word** sized signed result in the **second** operand. ...www.powermops.com/MopsManual/Assemblers/Chapter4.html - 49k - [Cached](#) - [Similar pages](#)Sega CD DevelopmentOn a Genesis, the **first** example would move a **longword** from the header of the ...The **second** example would store the number 1 in the **first** byte of Work RAM. ...www.retrodev.com/68ktut2.html - 9k - [Cached](#) - [Similar pages](#)Data processing device having a central processing unit and ...wherein when the **first** processing unit decodes the **second instruction** of the**second** type ... When **long word** data is input through the data bus 112 (in FIG. ...www.patentstorm.us/patents/6889240.html - 55k - [Cached](#) - [Similar pages](#)Did you mean to search for: **longsword** instruction first second



Result Page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [Next](#)

Free! Google Desktop Search: Search your own computer. [Download now.](#)

Find: emails - files - chats - web history - media - PDF

[Search within results](#) | [Language Tools](#) | [Search Tips](#) | [Dissatisfied?](#) [Help us improve](#)

[Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

©2005 Google

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S103	13	two adj variables same calculat\$4 same instruction\$2 <i>Rev all</i>	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/06/09 14:14
S102	38	generate near2 instruction same (first or second) near2 operand\$2	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/06/09 14:13
S101	32	compiler and long\$word and risc and add\$6 same subtract\$4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/06/09 14:12
S100	2	("same two variables") same instruction\$2	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/06/09 14:12
S99	2	"6718457".pn.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/06/09 13:00
S98	2	"6360312".pn.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/06/09 13:00
S97	2	"6718457".pn.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/06/09 12:29
S96	2	"20010032304"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/06/09 10:14
S95	4	"6076154".pn. 5922065.pn.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/06/09 10:14

S27	63	compil\$4 same combin\$4 near3 instructions	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/06/09 10:07
S94	70	combin\$4 near2 opcode\$2	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/06/09 09:42
S93	2	"6360312".pn.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/06/09 09:37